SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

16ITB201 – DESIGN AND ANALYSIS OF ALGORITHMS

II YEAR IV SEM

UNIT-I-Introduction

TOPIC: Fundamentals of the Analysis of Algorithm Efficiency – Asymptotic Notations

Prepared by
C.PARKAVI,AP/AIML

ASYMPTOTIC NOTATIONS AND ITS PROPERTIES

Subject :Design and Analysis of Algorithm
Unit :I

- Analysis Framework
- Asymptotic Notations and its properties
- Mathematical analysis for Recursive algorithms.
- Mathematical analysis for Nonrecursive algorithms.

Why Important

- Give a simple characterization of an algorithm's efficiency.

- Allow comparison of performances of various algorithms

Asymptotic Notations

- Big-oh Notation (O)

- Big-Omega Notation (Ω)

- Theta Notation (Θ)

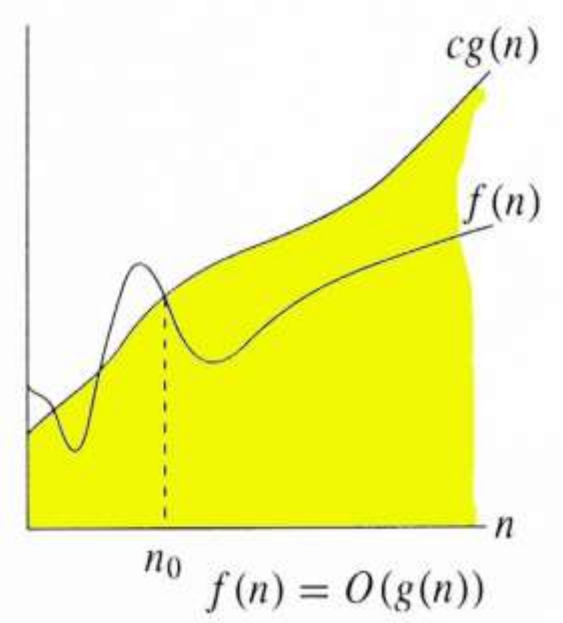
BIG-OH NOTATION (O)

- Gives the upper bound of algorithm's running time.
- Let f: N-> R be a function.

Then O(f) is the set of functions

O(f) = { g: N-> R | there exists a constant c and a natural number n_0 such that

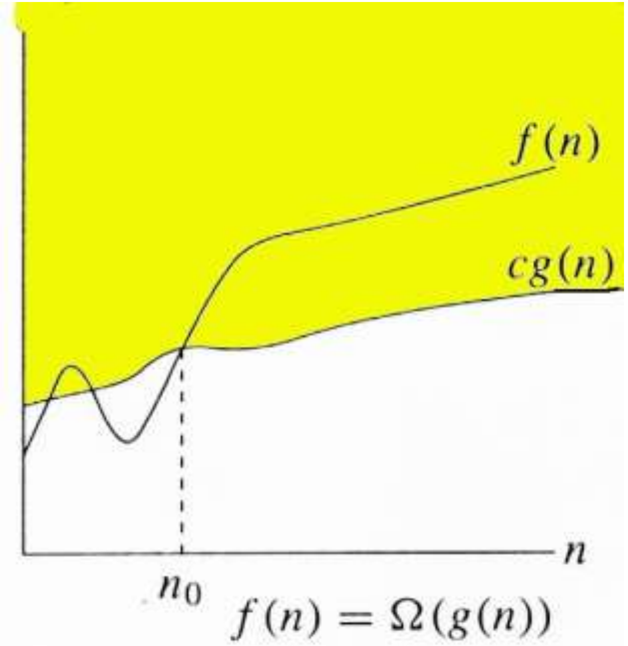
|g(n)| <= c|f(n)| for all n>= n_0 }



BIG-OMEGA NOTATION (Ω)

- Gives the lower bound of algorithm running time.
- Let f, g: N-> R be functions from the set of natural numbers to the set of real numbers.

We write g ∈ Ω(f) if and only if there exists some real number n_0 and a
positive real constant c such that
g(n)| >= c|f(n)|
for all n in N satisfying n>= n_0.



THETA NOTATION (Θ)

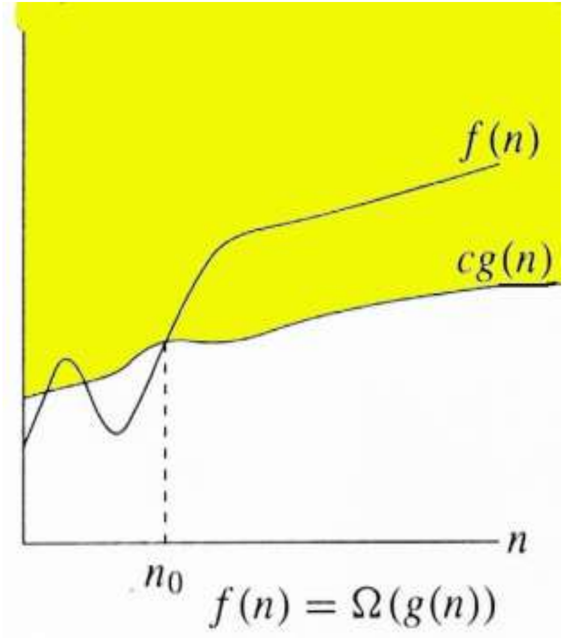
If f and g are functions from S to the real numbers, then we write $g \in \Theta(f)$ if and only if there exists some real number n_0 and positive

real constants C and C' such that

$C|f(n)| <= |g(n)| <= C'|f(n)|$

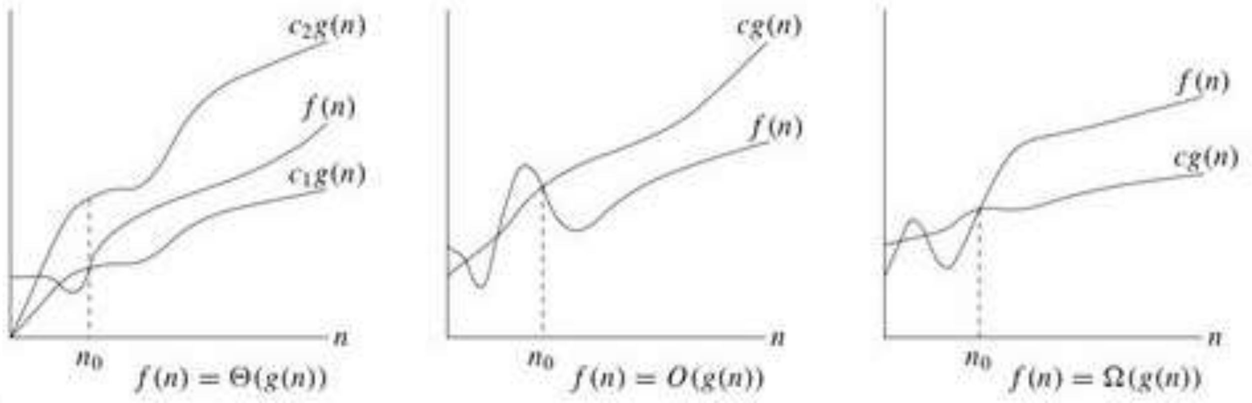
for all n in S satisfying $n >= n_0$.

Thus, $\Theta(f) = O(f) \cap \Omega(f)$



INTUITION ABOUT THE NOTATIONS

notation	intuition
O (Big-Oh)	$f(n)$ □ $g(n)$
□ (Big-Omega)	$f(n)$ □ $g(n)$
□ (Theta)	$f(n) = g(n)$

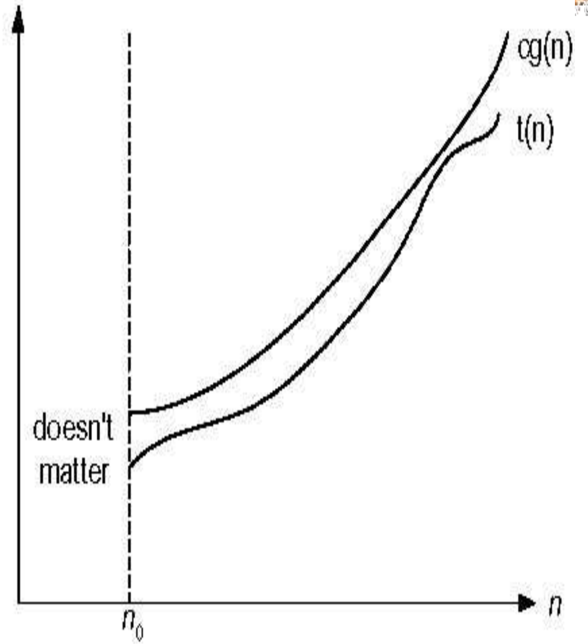


LITTLE OH NOTATION (O)

- **little-Oh Defn**:

$f(n) = o(g(n))$

- If for all positive constants c there exists an n_0 such that $f(n) < c \cdot g(n)$ for all n □ n_0.

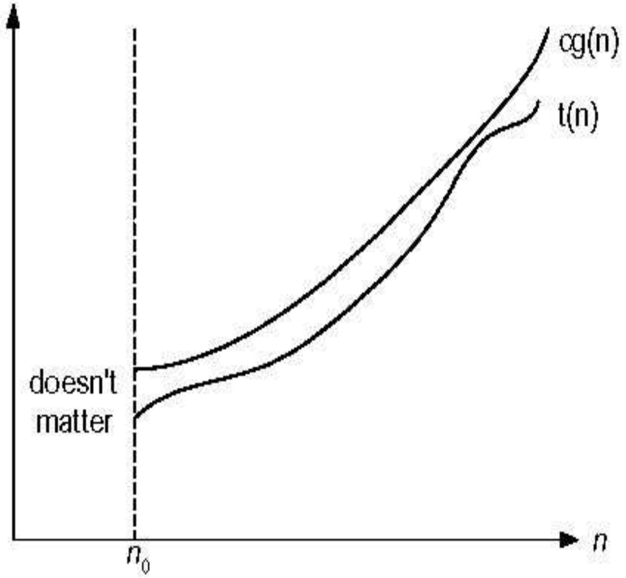


LITTLE OMEGA NOTATION (Ω)

- little-Omega Defn:

$f(n) = \omega(g(n))$

- If for all positive constants c there exists an n_0 such that $f(n) > c \cdot g(n)$ for all $n \geq n_0$.



TIME COMPLEXITY

Dependency of

- the time it takes to solve a problem
- as a function of the problem dimension/size

Examples:

- Sorting a list of length n
- Searching a list of length n
- Multiplying a $n \times n$ matrix by an $n \times 1$ vector

Time to solve problem might depend on data

- Average-case time
- Best-case time
  - data is well suited for algorithm (can't be counted on)
- Worst-case time
  - data is such that algorithm performs poorly (time-wise)

Worst-Case gives an upper bound as to how much time will be needed to solve any instance of the problem


Thank you!